iSelf: Towards Cold-Start Emotion Labeling using Transfer Learning with Smartphones

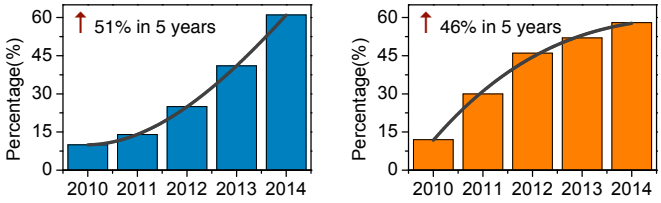
Boyuan Sun[1], Qiang Ma[1], Shanfeng Zhang[1,2], Kebin Liu[1], Yunhao Liu[1]
1 School of Software and TNList, Tsinghua University, Beijing, China
2 Department of Computer Science and Engineering, Hong Kong University of Science and Technology
{boyuan, maq, shanfeng, kebin, yunhao}@greenorbs.com

***Abstract*—To meet the demand of more intelligent automation services on smartphone, more and more applications are developed based on users' emotion and personality. It has been a consensus that a relationship exists between personal emotions and usage pattern of smartphone. Most of existing work studies this relationship by learning manually labeled samples collected from smartphone users. The manual labeling process, however, is time-consuming, labor-intensive and money-consuming. To address this issue, we propose iSelf, a system which provides a general service of automatic detection for user's emotions in cold-start conditions with smartphone. Using transfer learning technology, iSelf achieves high accuracy given only a few labeled samples. We also develop a hybrid public/personal inference engine and validation system, so as to make iSelf maintain continuous update. Through extensive experiments, the inferring accuracy is tested about 75% and can be improved increasingly through validation and update.**

I. Introduction

Nowadays, with the rapid development of mobile communication and sensor technology [1], the capability of smartphone has become very powerful. By utilizing various of functions, smartphone can bring us a lot of convenience. For example, the LBS can provide people accurate weather report or advertisement according to the location; the music player can store hundreds of music and we can listen to them anytime. These services can be implemented easily with the build-in equipments in smartphone. More and more applications, however, provide services based on human emotion or personality [2]. For instance, music player recommends users music list according to their current emotion state, and SNS introduces appropriate strangers according to people's personality [3]. As shown in Fig. I, for both Android OS and IOS, there have been more than 55% of such kind of emotion-related applications developed in online App Market in 2014.

Compared to manual input, a general service of automatic detection for user's emotion is much more practical. Several recent studies have investigated personal emotion and personality traits and have proved their relationship with usage patterns of smartphone. Most of them study this relationship by learning labeled samples collected from smartphone users. These training samples include two parts: 1) usage pattern of smartphone such as call logs, SMS logs, application usage logs and etc, 2) corresponding label of emotion. The labeled samples are leveraged to train an emotion classifier through some learning approaches, such as multi-linear regression [4], SVM [5, 6], C4.5 [7] and etc.



(a) In the Android App Market (b) In the IOS App Market

Fig. 1. The growth trend of emotion-related applications in the online App Market for Android OS and IOS.

Generally speaking, to guarantee a high accuracy, an efficient and valid classifier requires a big data set for training. For example, the authors in [4] leverage a training set including 32 iPhone users' daily usage report for more than two months. Note that, to obtain these samples is non-trivial. Although some background services are available for automatically collecting usage patterns, the labeling process must be done manually in the form of field study [4] or personality questionnaire [7], which is time-consuming, labor-intensive, and money-consuming. Moreover, even if we collect large amounts of labeled samples to train a classifier, unlike other recognition scenarios such as image recognition, there must be some feature spaces absent in the training set. For these human behaviors, even a strong classifier may become invalid and fail to infer users' emotions.

To address these issues, we propose iSelf, a system which can infer personal emotions automatically in cold-start conditions (i.e., with only a few labeled samples) on smartphone. iSelf collects three kinds of data: event data (e.g., calls and applications), sensor data(e.g., WiFi) and content data(e.g., SMS content). To measure the similarity between different usage patterns, iSelf conducts feature extraction for these raw data. By utilizing this feature similarity, iSelf realizes the feature-based transfer learning to infer emotions. To increase iSelf's inference accuracy, we validate the correctness of labeling results in two ways: automatic validation by overhearing emotion input and querying with minimal feedback using active learning. We also propose a hybrid public/personal inference engine which trains a personal classifier using the ground-truth data collected from the validation. In addition, iSelf updates the public/personal inference model to ensure continuous improvement of performance.

978-1-4799-8381-0/15/$31.00 ©2015 IEEE

The main contributions of this paper are summarized as follows:

- To the best of our knowledge, this work is the first to consider the cold-start problem of inferring personal emotions with smartphones.
- We design iSelf, a system which can infer personal emotions automatically with only a few labeled samples using transfer learning technology on smartphone.
- We propose a validation method to check the correctness of inference in two ways: automatic validation by overhearing emotion input and querying with minimal feedback using active learning.
- We conduct extensive experiments with more than 3600 samples of 10 participants during 30 days. iSelf achieves an inference accuracy of 75% and costs less than 2% of daily power consumption.

The rest of the paper is organized as follows. In Section II, we introduce the related work. In Section III, we describe the system overview. Section IV presents the detailed design of iSelf. We demonstrate the implementation, dataset collection, experiment results and discussions in Section V. The conclusion is presented in Section VI.

II. Related Work

This section surveys the existing methods [4–11] for emotion recognition. We classify the existing approaches for emotion recognition into two categories: 1) based on the relationship between human emotions and usage pattern of smartphone, 2) with the help of other equipments, such as video camera and facial features. Then we introduce the related work about transfer-learning [12–14].

A. Emotion Recognition through smartphone usage

A number of work [4–8] mainly focuses on recognizing emotions using smartphone usage patterns. MoodScope [4] proposes to infer mood based on how smartphone is used. The authors conducted a user study lasting for more than two months with 32 iPhone users. Similarly, other systems [5, 7, 6] utilize phone usage patterns to infer personality. Beside a large amount effort in collecting user patterns, they use questionnaires to label the data. In [7], the data is collected from smartphones of 83 individuals over a continuous period of 8 months, and TIPI [15] questionnaire is leveraged to determine the Big-five personality traits. In these approaches, data collection is time-consuming and labor-intensive. In contrast, iSelf conducts both data collection and labeling procedure automatically using only a few samples as trigger. In addition, iSelf increases detection accuracy continuously through validation and update.

B. Emotion Recognition with extra equipments

In this category, most of existing works utilize visual [16] and acoustic [17, 18] signals to extract speech, actions, and the facial features. For example, Mood Meter [19] counts smiles using video cameras. Others [20, 21] use physical signal such as skin conductance, heart rate, breath rate, blood pressure, and skin temperature. But these methods require additional hardware. In practice, these approaches are not suitable to our scenario due to the heavy sensing and computational burden. In contrast, iSelf only utilizes the usage patterns rather than sampling new signals. Meanwhile, iSelf avoids invasive image and audio data, such that it can run continuously in the background without compromising battery life.

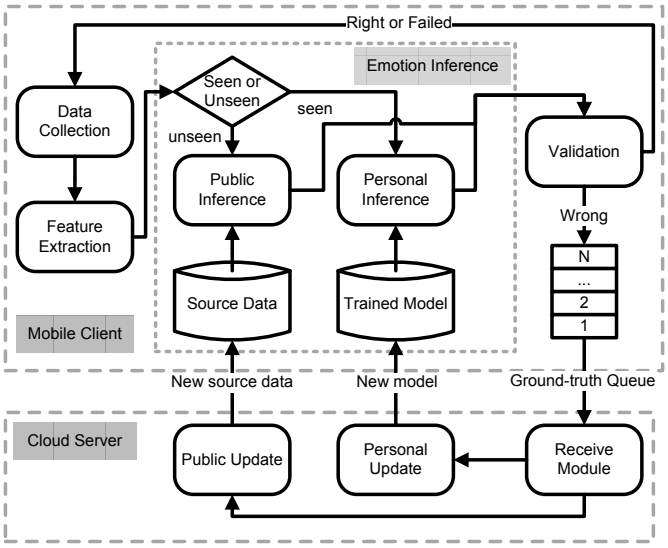


Fig. 2. System Architecture

C. Related work applied transfer learning

There are already some studies [12–14] about transfer learning. In [12], the author proposes a new dimensionality reduction method to find a latent space minimizing the distance between distributions of data in different domains. In this paper, the approach to transfer learning is verified by experiments in two real world applications: indoor WiFi localization and binary text classification. And in [13], the author proposes a transfer learning framework based on automatically learning a bridge between different sets of sensors to solve the activity recognition problem using transfer-learning technology.

III. System Overview

In this section, we present the system architecture of iSelf. As shown in Fig. 2, iSelf consists of two parts: mobile client and cloud server. On the mobile client, iSelf infers emotion using the user's smartphone usage patterns. On the server side, iSelf receives the ground-truth queue and updates the models. Finally, the mobile client downloads the new models. Next we describe each module in details.

A. Data Collection Module

Each data collection lasts for one hour. Generally, iSelf collects three kinds of data: event data, sensor data and content data. Event data include call logs, SMS logs and applications usage logs. Sensor data include activity states, location information, BT(BlueTooth) logs and WiFi signals. To save the power, iSelf collects the sensor data every 10 minutes

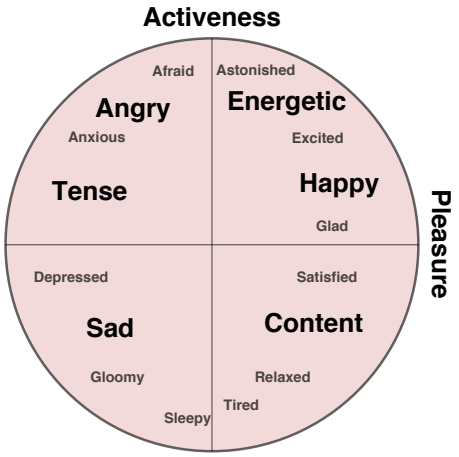


Fig. 3. Circumplex Emotion Model

and each collection lasts 5 minutes. Content data include SMS contents, online SNS contents and browser contents. For privacy issues, we keep the content data on the mobile client.

B. Feature Extraction Module

For event data, iSelf counts the number of each attribute such as the number of outgoing calls. For sensor data, iSelf transforms the multidimensional data into one dimension. Take acceleration date for example, iSelf transforms the three-dimensional raw data to user's activity state including silence, run and walk by analyzing the activity patterns. For content data, iSelf extracts the adjective, noun and emoticons and then analyses them using semantic analyze technology. Details are shown in Section IV-B.

C. Inference Module

We use the Circumplex emotion model [22] like MoodScope [4]. As shown in Fig. 3, the Circumplex model consists of two fundamental neurophysiological dimensions: a pleasure-displeasure dimension and an active-inactive dimension. Each emotion can be considered as a combination of these two dimensions [8]. We choose a set of standard and representative emotions: *sad, happy, angry, content, energetic, and tense*.

iSelf needs to be initialized with a small data set of labeled usage patterns. We collect data from 10 participants, covering all the six basic emotions. After feature extraction, we divide all the samples into six sets according to their labels.

There are two inference engines: public and personal. At the beginning, only public inference engine works, which is built with samples of all involved users by transfer learning. Personal inference is an inference engine constituted for a specific user. After collecting enough labeled data for each user, iSelf utilizes SVM [10] to train a personal model to infer his/her emotions.

D. Validation Module

iSelf validates the correctness of the result if either of two circumstances happens. One is that iSelf overhears users' inputs of their own emotions when they use some apps. The other is that iSelf measures the uncertainty and query user with minimal intervention. iSelf selects the first way preferentially. If iSelf gets the ground-truth label which is different from the inferred label, iSelf puts ground-truth to a queue and sends it to the cloud server. Otherwise, iSelf begins the next inference.

E. Update Module

This module is deployed on the cloud server. When the cloud server receives the ground-truth queue, iSelf puts it to corresponding emotion set in source domain and re-trains the personal classifier to get a stronger emotion model. When monitoring the WiFi environment and idle state of smartphone, e.g., at night when the user is sleeping, iSelf downloads the new models.

IV. SYSTEM DESIGN

In this section, we present the design of iSelf in details. At first, we only have a few samples collected from 10 participants. Then we utilize the transfer learning technique to automatically label the input usage pattern. After collecting enough labeled data, iSelf trains a personal classifier using the collected data to help recognizing emotions. Before training, iSelf validates the correctness of inferred emotions. Finally, iSelf increases detecting accuracy by updating models.

A. Data Collection

We build iSelf's input feature vector using the usage records collected by the logger. It has been suggested by the literature that emotion is strongly related with the social interaction [11] and daily activity [9] of a person. Our collected data belong to both of them.

Feature Vector: Every data collection lasts one hour. Here we ignore the instantaneous emotion persisting for only seconds. Emotion inference is based on the data collected in this one hour. We collect three kinds of data: event data, sensor data and content data. At beginning, iSelf explores the user's smartphone usage history to make sure the unique contacts by calculating the number and duration of the communication. As a matter of fact, the confidence of the inference through these contacts is very high. iSelf regards the top 10 call numbers and top 20 SMS numbers as the unique contacts. We design a background service to collect these three kinds of data. Table I shows the data types required in detail. Here, iSelf collects sensor data every 10 minutes and every collection lasts 5 minutes to save the battery power. These three kinds of data are combined as the input feature vector.

B. Feature Extraction

Before combining these three kinds of data, an important step is feature extraction. For event data, we count the number of each attribute. For sensor data, we handle the data of accelerometer and location(GPS). For accelerometer, we extract the feature of the raw data stream to get the state of people including run, walk and silence. We approximate the force exerted by people as follows:

TABLE I
FEATURE VECTOR

Date Style	Date Type	Usage Cue
Event Data	Calls	No. of outgoing calls No. of incoming calls duration of each call No. of top 10 contacts called No. of top 10 contacts who called No. of missed calls
	SMS	No. of SMS received No. of SMS sent No. of Top 20 contacts received from No. of Top 20 contacts sent to
	Application	No. of uses of Office Apps No. of uses of Email Apps No. of uses of Video/Music Apps No. of uses of Chat Apps: Wechat,etc No. of uses of SMS App No. of uses of Camera App No. of uses of Map app No. of uses of Games The time of each app used
Sensor Data	Accelerometer	X,Y,Z Accelerator
	GPS	Altitude, latitude, longitude The time of locations collected
	Bluetooth	No. of BT IDs BT IDs for more than 3 time slots Max. time for a BT ID seen
	Wifi	No.of Wifi signals in each time slot
Content Data	SMS content	Average length of SMS Content of each SMS
	Online SNS	Key Words Expression tags
	Browser	Browser Search content Browser Bookmarks content

$HF = \sqrt{Accel_x{}^2 + Accel_y{}^2 + Accel_z{}^2} - G(Gravity)$. We define two thresholds, *AccelThreshold1* and *AccelThreshold2* where *AccelThreshold1* < *AccelThreshold2*. We assume the state is run if the HF is greater than *AccelThreshold2*. The state is walk if HF is greater than *AccelThreshold1* and less than *AccelThreshold2*. The state is silence when the HF is less than *AccelThreshold1*. Then we change the raw data stream to the state stream. *For location*, we cluster our time-series of location data through the DBSCAN, which allows us to get the visited locations. For content data, we divide the content into textual data and emoticons. We extract the adjective and noun from the text and convert the emoticons into corresponding emotional text. Table II shows the partial converting rules.

Finally we classify them into three categories according to data type. They are statistical data(SD), stream flow data(SF) and textual data(TD) corresponding to event data, sensor data and content data, respectively.

C. Automatically Label

We adopt the transfer learning technology [13] to realize the automatic labeling. To formalize the labeling problem, we define the labeled samples collected from 10 participants in the form of (x_s, y_s), where x_s means labeled feature vector, y_s means emotion label and s means source domain. Then we define the new unlabeled input feature vector as x_t, where t means target domain. What we want to know is the corresponding y_t. y_s and y_t belong to the same label space L

TABLE II
CONVERT RULE

happy	$(\hat{}\_\hat{}) \quad (*\hat{}\_\_\hat{}*) \quad (\hat{}o\hat{}) \quad (\hat{}.\hat{})$
sad	$(T\_T) \quad (T.T) \quad (T\hat{}T) \quad ()$
angry	$(>\hat{}<) \quad (>\_<)$
content	$(\sim\hat{}\sim) \quad <(\sim)\sim> \quad <(-v-)>$
tense	$o\_O \quad (@\hat{}@) \quad (\hat{};;;;;$
energetic	N/A

which includes {sad, happy, angry, content, energetic, tense}. But x_s and x_t are not in the same feature space because different people have different smartphone usage patterns under the same emotion. Our final goal is to estimate $p(y_t|x_t)$. By transfer learning [13], we have:

$$p(y_t|x_t) = \sum_{c^{(i)} \in L} p(y_t|\mathbf{c}) p(\mathbf{c}|x_t) \tag{1}$$

From the above equation, the automatic labeling takes two steps. First, to estimate every $p(\mathbf{c}|x_t)$ where $\mathbf{c}$ is labeled using the source domain label space. In other words, we aim to use the source domain label space to label the target domain feature space x_t. The target domain feature space may be much different from the source domain feature space and unseen before. So first of all, we need to transfer across different feature spaces. What follows is to calculate $p(y_t|\hat{\mathbf{c}})$.

Transfer Across Feature Vectors: As discussed above, we need to transfer knowledge between different feature vectors and then estimate $p(y_t|\mathbf{c})$. For each feature vector x_s in the source domain S, x_s is represented by features f_s. For the new reading vector x_t in the target domain T, then the features of x_t is represented as f_t. f_s is the labeled samples from the 10 participants, while f_t is the unseen smartphone usage patterns from other people. f_s and f_t are very different because people have different usage habits and even for one person, the usage patterns vary at different time. So what we should do is to build a bridge between f_s and f_t. We use a framework similar to translated learning [14]. The challenge now is to find a *translator* $T(f_s, f_t) \propto p(f_t|f_s)$. Due to $p(f_t|f_s) = p(f_t, f_s)/p(f_s)$, we focus on $p(f_t, f_s)$. We have:

$$p(f_t, f_s) = \int_{\mathcal{X}_s} p(f_t|x_s, f_s) p(f_s|x_s) p(x_s) d_{x_s} \tag{2}$$

For x_s, f_s and f_t are independent, the equation 2 can be rewritten as:

$$p(f_t, f_s) = \int_{\mathcal{X}_s} p(f_t|x_s) p(f_s|x_s) p(x_s) d_{x_s} \tag{3}$$

$$= \int_{\mathcal{X}_s} p(f_t, x_s) p(f_s|x_s) d_{x_s} \tag{4}$$

To measure $p(f_t, f_s)$, $p(f_t, x_s)$ is necessary. In other words, we need to measure the relationship between the input feature vector and feature vectors x_s from source domain. Because the feature vector has three data categories, we measure $p(f_t, x_s)$

according to different category. Then we can convert $p(f_t, x_s)$ into the following equation:

$$p({f_t}^{SD}, {x_s}^{SD}) = \frac{\sum_{{f_t}^{(i)} \in SD_t} \sum_{{x_s}^{(j)} \in SD_s} p({f_t}^{(i)}, {x_s}^{(j)})}{|SD_t||SD_s|} \quad (5)$$

$$p({f_t}^{SF}, {x_s}^{SF}) = \frac{\sum_{{f_t}^{(i)} \in SF_t} \sum_{{x_s}^{(j)} \in SF_s} p({f_t}^{(i)}, {x_s}^{(j)})}{|SF_t||SF_s|} \quad (6)$$

$$p({f_t}^{TD}, {x_s}^{TD}) = \frac{\sum_{{f_t}^{(i)} \in TD_t} \sum_{{x_s}^{(j)} \in TD_s} p({f_t}^{(i)}, {x_s}^{(j)})}{|TD_t||TD_s|} \quad (7)$$

$$p(f_t, x_s) \approx \frac{p({f_t}^{SD}, {x_s}^{SD}) + p({f_t}^{SF}, {x_s}^{SF}) + p({f_t}^{TD}, {x_s}^{TD})}{3} \quad (8)$$

where SD_t is the feature set belongs to statistical data type of target domain and SD_s means the same data type of the source domain. SF_s, SF_t, TD_S, TD_t are defined similarly. Instead of measuring the $p(f_t, x_s)$ directly, we calculate $p({f_t}^{SD}, {x_s}^{SD})$, $p({f_t}^{SF}, {x_s}^{SF})$, and $p({f_t}^{TD}, {x_s}^{TD})$, respectively. We use Jeffrey's J-divergence [23](the symmetric version of KL-divergence) to approximate $p({f_t}^{SD}, {x_s}^{SD})$, Dynamic Time Warping [24] to approximate $p({f_t}^{SF}, {x_s}^{SF})$ and Cosine similarity to approximate $p({f_t}^{TD}, {x_s}^{TD})$.

To measure $p({f_t}^{SD}, {x_s}^{SD})$, we estimate each probability distribution in calls, SMS and applications. Take calls as an example, we simply estimate the probability $p(OutgoingCalls)$ as $\frac{\mathcal{NOC}}{\mathcal{NC}}$ where $\mathcal{NOC}$ means the number of outgoing calls and $\mathcal{NC}$ means the number of calls. Similarly, we can estimate $p(IncomingCalls)$, $p(Top10ContactsCalled)$ and $p(Top10ContactsWhoCalled)$. For SMS and applications, we adopt the same method to estimate their probability distributions. We define the estimated distribution as $\mathcal{A}$ and we wish to find a close distribution $\mathcal{B}$ in the source domain. Since $D_{KL}(\mathcal{A}||\mathcal{B})$ is not equal to $D_{KL}(\mathcal{B}||\mathcal{A})$. We use $D_{KL}(\mathcal{A}||\mathcal{B})$+$D_{KL}(\mathcal{B}||\mathcal{A})$ instead which is undoubtedly symmetric. As definition of J-divergence, the more similar $\mathcal{A}$ and $\mathcal{B}$ are, the lower the value of $D_{KL}(\mathcal{A}||\mathcal{B})$+$D_{KL}(\mathcal{B}||\mathcal{A})$ is. So we only consider distribution pairs at low divergence values.

To measure $p({f_t}^{SF}, {x_s}^{SF})$, we first normalize all the stream-flow data readings into the range of [0,1]. Then we consider the sampling rates of different data types may be different such as the activity state stream and the WiFi signals. To solve this problem, we choose a distance metric that can take different sampling rates into account. Now given two series of sensor readings of only one dimension such as activity state stream and WiFi signals: M and N of length m and n, we use dynamic time warping(DTW) [24] to measure the similarity of M and N. DTW uses dynamic programming to calculate the matching cost of two time series and find the optimal path. The optimal path from (1,1) to (i,j) must come from the optimal paths from (1,1) to the three predecessor candidates include (i-1,j), (i-1,j-1) and (i,j-1). Then the matching cost from (1,1) to (i,j) is distance at (i,j) add the smallest one of these three candidates. The time and space complexity of DTW are both $O(M * N)$. The smaller the calculated distance is, the more similar M and N are. So we only select the low DTW values.

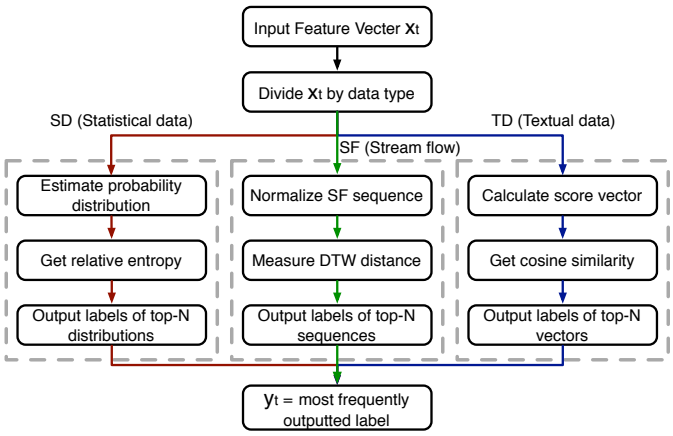


Fig. 4. Automatic Labeling

To measure $p({f_t}^{TD}, {x_s}^{TD})$, we extract the emoticons, adjectives and nouns and convert them into scores between [-1,1] using SentiWordNet [25]. We construct a vector $W = \{W_1, W_2, W_3, W_4, W_5\}$ where W_i means the number of words with scores belonging to [-1+(i-1)*0.4, -1+i*0.4]. Similarity between ${f_t}^{TD}$ and ${x_s}^{TD}$ is calculated using Equation 9.

$$similarity = cos(\theta) = \frac{W_t \cdot W_s}{||W_t|| \cdot ||W_s||} \quad (9)$$

where W_t and W_s are the vectors of ${f_t}^{TD}$ and ${x_s}^{TD}$.

After estimating $p(\mathbf{c}|x_t)$, we calculate $p(y_t|\mathbf{c})$. If $y_t = \mathbf{c}$, then $p(y_t|\mathbf{c}) = 1$ and if $y_t \neq \mathbf{c}$, then $p(y_t|\mathbf{c}) = 0$. Then we finish the automatic labeling. Figure 4 shows the steps of automatic labeling.

D. Validation

There are two ways for validating the correction of inferred emotion: 1) **Automatic Validation**: iSelf overhears users' input of their own emotions; 2) **Query with Minimal Feedback**: iSelf utilizes active learning method to realize minimal user feedback. There is no user intervention in the first way at all.

Automatic Validation: iSelf overhears users' inputs of their own emotions. When users use some applications such as music player and SNS, they may input their emotions as a query(e.g.Moodagent) or sharing with others(e.g.Facebook), which can be overheard by iSelf as the ground-truth. In each slot, iSelf collects a series of user's input emotions as $E = \{e_1, e_2, e_3, ..., e_n\}$. Then iSelf calculates the similarity $S(e_i|y)$, where $e_i \in E$ and $y \in \{sad, happy, angry, content, energetic, tense\}$. Since users input may be different from our basic emotions but have the same semantic meaning, we utilize SentiWordNet [25] to calculate the scores of these emotion words and measure the similarity by comparing the scores. iSelf takes the basic emotion with the most occurrences as the ground-truth.

Reinforce Recognition Using Minimal Feedback: When no inputs about emotion are overheard, iSelf asks users to

label the usage pattern. Obviously, it is impractical to query a user every time. The more frequently iSelf asks users, the more intrusive the system is. To address this issue, we use the idea of active learning to measure the uncertainty of a sample through calculating maximum entropy. The equation $E_m(Y|x_t) = -\sum_{y_t} p_m(y_t|x_t) log p_m(y_t|x_t)$ means the uncertainty the classifier is about the value of label Y given a feature vector x_t and classifier model m. We define a threshold e, and iSelf asks user for a ground-truth label when $E_m(Y|x_t)$ is not less than e.

The complete algorithm is shown in Algorithm 1. If the inferred emotion is wrong, iSelf puts the ground-truth into a queue and sends it to cloud server for updating.

Algorithm 1 Validation

Input: Collected feature vector x_t; Inferred emotion label y_t; An initial classifier model m; Defined threshold e;
Output: Ground-truth emotion E_g

```
Define a emotion set E
while iSelf Service is running do
    if iSelf overhears a user's input about emotion then
        iSelf put the emotion e_i into E
    if size of E is equal to 0 then
        E_m(Y|x_t) = -Σ_{y_t} p_m(y_t|x_t) log p_m(y_t|x_t)
        if E_m(Y|x_t) ≥ e then
            E_g ← queryForLabel(x_t)
        else
            E_g ← NULL
    else
        Calculate Similarity S(e_i|y)
        Map E to the six basic emotions
        Take the emotion with most occurrences as E_g
return E_g
```

E. Hybrid Public/Personal Inference Engine

While utilizing transfer learning method to recognize a user's emotion, there are certain amounts of useful information in the target feature space that we do not want to discard. A period of time later, e.g. one week, for a specific user, many usage patterns with ground-truth label are collected through validating. With the help of these records, iSelf can improve detecting accuracy for this person. Thus we propose a hybrid public/personal inference method. Public inference is the inference engine utilizing transfer learning method to infer emotions for everyone. Personal inference is an inference engine constituted for a specific user.

To build personal inference engine, our idea is to save the previous labeled usage patterns and train a personal classifier. To infer a specific user's emotions, if we know that a feature vector belongs to the same feature space with the classifier, we can directly apply the personal classifier. Otherwise, we use transfer learning method.

After the training sets are constructed, a binary classifier is trained for each basic emotion. If the emotion only have a few samples or does not have any sample, iSelf does not train a classifier for this emotion. Compared with various classifiers, we select the Support Vector Machine(SVM) classifier [10]. SVM searches the hyperplane $\mathbf{w}^T\mathbf{x}_i + b = 0$ that maximizes the margin between points from different labels by optimizing the following Quadratic Programming equation:

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i \tag{10}$$

$$s.t. \quad a_i(w^T\mathbf{x}_i + b) \geq 1 - \xi_i \tag{11}$$

$$\xi_i \geq 0, \quad \forall i \tag{12}$$

where $\mathbf{x}_i$ and a_i are the feature vector and label value for the i-th training sample; $\mathbf{w}$ and b controls the offset and orientation of the hyperplane; C refers to a regularization term used to control the overfitting and the false classification tolerance ξ_i for each sample.

After we train a classifier for each basic emotion. Now we face a problem is: How can iSelf know if a feature vector belongs to a seen feature space? We develop an "anomaly" detector. If a feature vector is from a seen feature space, it is similar to the samples in the personal training set. Otherwise, the feature vector is different. To detect an "anomaly", we first train an unseen feature space detector using the one-class SVM classifier [10]. All the usage patterns of a user collected by iSelf as the positive samples(no negative samples) are trained to get a personal classifier for this user to detect if the feature space is unseen. The complete algorithm of hybrid public/personal inference is shown in Algorithm 2.

Algorithm 2 Hybrid Public/Personal Inference

Input: feature vector x_t
Output: inferred emotion y_t

```
isUnseen ← UnseenFeatureSpaceDetection(x_t);
if isUnseen = true then
    y_t ← Automatic-Labeling;
else
    y_t ← PersonalSVMClassifier(x_t);
return y_t
```

F. Update

After validating, iSelf sends the ground-truth queue to cloud server and updates the public source domain and the personal classifier. First, iSelf puts the usage patterns to the source domain sets. It increases the possibility that classify the similar usage patterns to the truth emotion. Second, iSelf adds the queue to the personal training samples to re-train a stronger personal classifier. For this user, the personal SVM classifier can be more accurate next time. After updating these two parts, iSelf sends the new models back to mobile client.

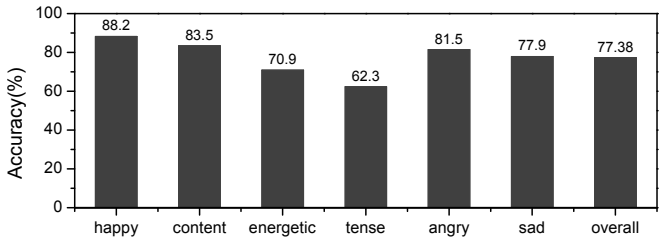


Fig. 5. Inference Accuracy

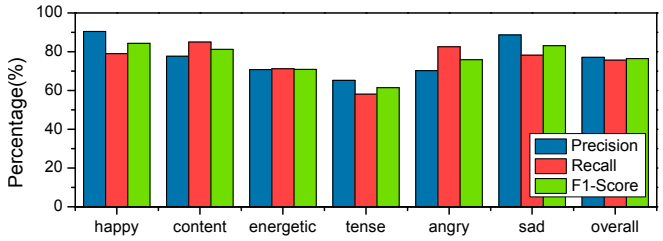


Fig. 6. Precision, Recall and F1-Score

V. Evaluation

A. System Implementation

We have implemented iSelf system on Samsung GALAXY Note1 which has a three-dimensional accelerometer, WiFi, Bluetooth, GPS, and other basic equipments. The system runs on the Android OS2.3.3. We implemented the code for data collection, feature extraction, automatically label, validation, hybrid classification and update. We also build a cloud server on Sina App Engine in Java. SVM classifier is implemented using the LibSVM library.

B. Datasets

We collect participants' mobile usage patterns including event data, sensor data and content data for one hour every time, and ask them to label the corresponding emotions. 10 participants (4 females and 6 males) install the service and collect about 3600 records for 30 days. They are undergraduates, postgraduate and common workers and their ages range from 20 to 40. We regard these labeled usage patterns as the source domain.

C. Evaluation Methodology

We use the leave-M-out validation method. We have N persons, and each time we take M persons as target and the rest $N-M$ persons as source. We test all $\binom{N}{M}$ target/source combinations. Three metrics are evaluated. They are defined as follows:

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN} \tag{13}$$

$$F1-score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{14}$$

where *TP*, *FP*, *TN* and *FN* means true positive, false positive, true negative and false negative, respectively. *Precision* means the percentage of correct emotion inference made by the system. *Recall* is the percentage of an emotion detected. *F1-score* means the combination of them. For iSelf's inference *accuracy*, it is calculated as the number of correct inferred emotions divided by the number of all the test samples.

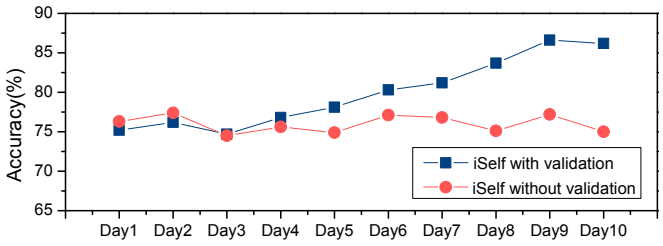


Fig. 7. Accuracy Variation of Validation

D. System Performance

Inference Accuracy: We set $M = 2$. Figure 5 shows the accuracy of iSelf and Fig. 6 shows the corresponding precision, recall and f1-score. The overall accuracy is 77.4% and F1-score is 76.4% over all emotions. Three emotions including happy, content and sad reach a promising accuracy and F1-score of over 80%. The tense has the minimal accuracy of about 60% and happy is the highest one about 90%. This results support our theory that unseen feature space can be labeled automatically through transfer learning technology.

We can draw from the experimental result is that misclassification usually happens when two feature spaces of one emotion have very large difference. We discover that usage patterns vary much when people are tense and this leads to low inference accuracy.

Inference accuracy variation of a user: As time goes on, the accuracy of a user increases due to the validation. Through validation, the specific SVM classifier becomes more robust with just a few days. Figure 7 shows the accuracy variation of a user in 10 days and the accuracy without validation. In the first 4 days, accuracies are close. From the fifth day, accuracy with validation increases. This is because there are not enough samples to train a personal SVM classifier in the first 4 days.

System Overhead: Since emotion model training and updating are conducted offline on the cloud server, we mainly consider the power consumption of emotion inference.

iSelf occupies only 4.6M storage when it is running. Then we measure the energy consumption of data collection, pre-processing, inference, and interaction with server. Meanwhile, we calculate each computation time. We also test the size of files uploaded to and downloaded from the server. All the results are displayed in Table III.

We obtain the system power consumption of GALAXY Note1 using a resistor put in series with the battery. As we can see, the power consumption is less than 500 mW during data collection, pre-processing, and inference. In a whole day, iSelf costs less than 2% of a phone's total power consumption.

Although iSelf needs to open sensors and monitor the running events and input content during data collection, it

TABLE III
SYSTEM OVERHEAD

Data Collection(once/hour)	
Power Consumption	110 mW
Computation Time	109 ms
Pre-processing(once/hour)	
Power Consumption	122 mW
Computation Time	1.7 s
Inference(once/hour)	
Power Consumption	246 mW
Computation Time	464 ms
Interaction(once/day)	
Data Upload	2MB
Data Download	10KB
Power Consumption	3036 mW
Time to send/receive	2.3 s

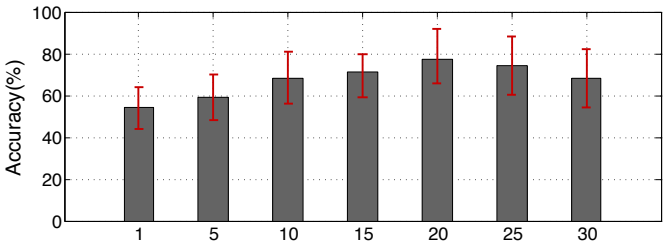


Fig. 8. Impact Of N in automatic labeling

conducts this every 10 minutes. So in fact, data collection costs a little energy. Feature Extraction is lightweight, only involving simple data transformation and DBSCAN clustering [26, 27]. Inference contains hybrid public/personal emotion detection and validation. Since transferring needs some computation, inference costs more power which reaches 246 mW. Finally, iSelf must upload the entire usage log and download a new specific SVM model.

E. Impact of Different Parameters:

Impact of Parameter N in automatic labeling: We set $M = 2$ and report the detecting accuracy of iSelf by varying the parameter N in Fig. 4. We select the top-N similar SD distributions and top-N minimum SF DTW scores, as well as top-N similar TD contents, a total of $3N$ input feature vectors and corresponding labels. Our result in Fig. 8 shows that the accuracy increases with N increasing. This is because more candidate labels are taken into account and thus we can consider more "probabilities". However, when N is larger than 25, the accuracy drops slightly due to the influence of noises.

Impact of M: Let $N = 10$. We set M from 2 to 9, and check the impact of number of unseen persons. The results are shown in Fig. 9. As we can see, when the number of seen persons is equal or greater than 7, the accuracy stays constant. System can maintain an accuracy of over 65% when there are only 6 seen persons. When there is only one seen person, the detecting accuracy drops to 11.2%. Overall, iSelf can achieve

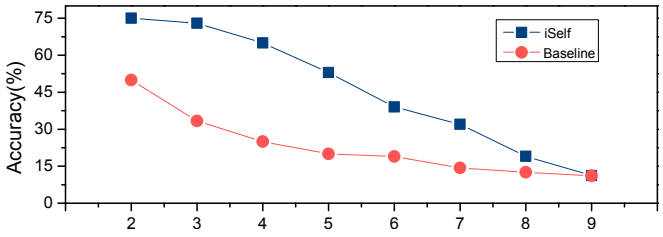


Fig. 9. Impact of M, the number of testing persons

approximately 20-30% better accuracy than the baseline if four or more persons are seen.

F. Case Study: Analyze Inherent Reason of results

Same emotion can cause similar SD or SF: It is obvious that contents data sometimes can express explicit emotions. For example, a user sends a message like "I was punished by my teacher this afternoon. What a sad day!". We can infer that this user is angry or sad. Such messages can help iSelf improve recognizing accuracy. But most time, iSelf only has SD or SF or SD+SF. Can iSelf infer emotions correctly without TD information? Through our experiment, we discover that if a user expresses an emotion with the (SD or SF)+TD, the user expresses a similar emotion if he/she only has the similar SD or SF. So we conclude that the same emotion can cause similar SD or SF with over 65% probability.

Interesting Things: We find most people have more calls or SMS when they are happy or sad. When they are happy, people are usually outdoor(WiFi) or crowded by others(Bluetooth). But when they are sad, people are always indoor or sole. We consider people prefer to stay alone when they are sad. Also we discover that people are usually content or happy when they use the camera application. People are likely to be angry or tense when they use the music player application. Then people are always energetic when the activity state is running or more locations are visited.

VI. CONCLUSION

In this paper, we have shown the design, implementation, and evaluation of iSelf, a system that automatically infers emotions while the feature space is unseen before. Previous work can only infer emotions with the seen input feature space leading to time-consuming, labor-intensive and money-consuming collection and labeling. iSelf leverages transfer learning technology to infer emotions though the input feature space is unseen. We only need to collect a little labeled data from several people and it saves time, labor and money. Also, previous work gets lower accuracy if the input feature space is unseen and iSelf solves this problem. Validation is developed in two ways to improve the performance with minimal user feedback. iSelf achieved up to an average of 75% inference accuracy on the unseen feature space.

VII. Acknowlegment

Many thanks to the anonymous reviews for their constructive comments. The research of Qiang Ma is supported in part by NSF China Project under Grant No. 61472219. The research of Shanfeng Zhang and Kebin Liu is supported in part by NSF China Projects under Grant No. 61472218 and 61472337. The research of Yunhao Liu is supported in part by the NSF China Distinguished Young Scholars Program under Grant No. 61125202.

References

[1] Z. Li, M. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2011.

[2] Z. Liu, Y. Feng, and B. Li, "Socialize spontaneously with mobile applications," in *Proceedings of IEEE INFOCOM*, 2012.

[3] Y. Li and J. C. S. Lui, "Friends or foes: Detecting dishonest recommenders in online social networks," in *Proceedings of IEEE ICCCN*, 2011.

[4] R. LiKamWa, Y. Liu, N. D. Lane, and L. Zhong, "Moodscope: building a mood sensor from smartphone usage patterns," in *Proceedings of ACM MobiSys*, 2013.

[5] G. Chittaranjan, J. Blom, and D. Gatica-Perez, "Mining large-scale smartphone data for personality studies," *Personal and Ubiquitous Computing*, vol. 17, no. 3, pp. 433–450, 2013.

[6] Y.-A. de Montjoye, J. Quoidbach, F. Robic, and A. S. Pentland, "Predicting personality using novel mobile phone-based metrics," in *Proceedings of LNCS SBP*, 2013.

[7] G. Chittaranjan, J. Blom, and D. Gatica-Perez, "Who's who with big-five: Analyzing and classifying personality traits with smartphones," in *Proceedings of IEEE ISWC*, 2011.

[8] K. Church, E. E. Hoggan, and N. Oliver, "A study of mobile mood awareness and communication through mobimood," in *Proceedings of ACM NordiCHI*, 2010.

[9] L. A. Clark and D. Watson, "Mood and the mundane: relations between daily life events and self-reported mood." *Journal of personality and social psychology*, vol. 54, no. 2, p. 296, 1988.

[10] L. M. Manevitz and M. Yousef, "One-class svms for document classification," *the Journal of machine Learning research*, vol. 2, pp. 139–154, 2002.

[11] J. P. Forgas, G. H. Bower, and S. E. Krantz, "The influence of mood on perceptions of social interactions," *Journal of Experimental Social Psychology*, vol. 20, no. 6, pp. 497–513, 1984.

[12] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, 2008, pp. 677–682.

[13] D. H. Hu and Q. Yang, "Transfer learning for activity recognition via sensor mapping," in *Proceedings of ACM IJCAI*, 2011.

[14] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proceedings of ACM NIPS*, 2008.

[15] S. D. Gosling, P. J. Rentfrow, and W. B. Swann Jr, "A very brief measure of the big-five personality domains," *Journal of Research in personality*, vol. 37, no. 6, pp. 504–528, 2003.

[16] G. Wang, M. Z. A. Bhuiyan, J. Cao, and J. Wu, "Detecting movements of a target using face tracking in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 939–949, 2014.

[17] C. M. Lee and S. S. Narayanan, "Toward detecting emotions in spoken dialogs," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 2, pp. 293–303, 2005.

[18] B. Schuller, R. J. Villar, G. Rigoll, and M. K. Lang, "Meta-classifiers in acoustic and linguistic feature fusion-based affect recognition." in *Proceedings of IEEE ICASSP*, 2005.

[19] J. Hernandez, M. E. Hoque, W. Drevo, and R. W. Picard, "Mood meter: counting smiles in the wild," in *Proceedings of ACM UbiComp*, 2012.

[20] A. Gluhak, M. Presser, L. Zhu, S. Esfandiyari, and S. Kupschick, "Towards mood based mobile services and applications," in *Proceedings of ACM EuroSSC*, 2007.

[21] J. F. Cohn, "Foundations of human computing: facial expression and emotion," in *Proceedings of ACM ICMI*, 2006.

[22] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[23] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.

[24] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proceedings of ACM KDD*, 2000.

[25] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining." in *Proceedings of ELRA LREC*, 2010.

[26] S. Liu, Y. Liu, L. M. Ni, J. Fan, and M. Li, "Towards mobility-based clustering," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, 2010, pp. 919–928.

[27] Y. Yu, Z. Chen, B. Cao, W. Dong, Y. Guo, and J. Cao, "Mobsafe: cloud computing based forensic analysis for massive mobile applications using data mining," *Tsinghua Science and Technology*, vol. 18, no. 4, pp. 418–427, 2013.